SOFTWARE FOR AUTOMATIC PROGRAMMING OF NUMERICALLY CONTROLLED DRILLING MACHINES

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

BY
P. VIJAYA SHANKAR NAYAK

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

AUGUST 1972

CERTIFICATE

This is to certify that the thesis entitled, "SOFTWARE FOR AUTOMATIC PROGRAMMING OF NUMERICALLY CONTROLLED DRILLING MACHINES" is a record of the work carried out under my supervision and that it has not been submitted elsewhere for a degree.

[illegible]

H.N. Mahabala
Associate Professor of
Electrical Engineering
and
Head, Computer Centre
Indian Institute of Technology, Kanpur

Kanpur
August, 1972

POST GRADUATE OFFICE
This thesis has been approved
for the award of the Degree of
Master of Technology (M.Tech.)
in accordance with the
regulations of the Indian
Institute of Technology Kanpur
Dated. 30.8.72 [illegible]

EE-1972-M-NAY-SOF

V
JUNE '76

29 SEP 1972
I. I. T. KANPUR
CENTRAL LIBRARY
Acc. No. A 21143

Thesis
621.9520285425
N231

ACKNOWLEDGEMENTS

I am deeply indebted to Dr. H.N. Mahabala, Associate Professor in Electrical Engineering and Head of the Computer Centre, Indian Institute of Technology, Kanpur for his valuable guidance and words of encouragement which are a continuous source of inspiration. I am very grateful to Dr. T. Radhakrishnan, Lecturer in Electrical Engineering and Computer Sciences, IIT Kanpur, for his deep sense of understanding and keen interest in my work. My thanks are also due to Messrs Heavy Electricals (India) Limited, Bhopal for making this interesting project possible.

Kanpur
August 1972

- P. Vijaya Shankar Nayak

SYNOPSIS

DRILLPAL is a special purpose software package developed for computer production of control paper tapes for the Numerically Controlled multispindle drilling machines INOCENTI and LAHR of Messrs Heavy Electricals Ltd., Bhopal.

Manual preparation of control tapes is not only tedious and time consuming but also is very error prone. With the computer, one can attempt to minimise the number of drill movements also.

The package provides an elegant and easy to learn partprogramming language with which the partprogrammer is able to describe the pattern of holes to be drilled and also specify associated technical details such as tool description, work size etc.

The package functionally consists of an interpreter, a simulator and a post-processor. The interpreter processes the partprogram and produces an image in core, of the pattern of holes described. The drilling machine is simulated and control information in appropriate form is produced by the simulator. This inturn is converted by the post-processor to a paper tape of the required format.

By the use of the simulator one can follow the actual progress of the job. It is also easy to incorporate new optimization techniques for optimizing the drilling sequence.

CONTENTS

CHAPTER 1

INTRODUCTION

This chapter gives a brief introduction to Numerical Controlled machine tools with special emphasis on NC drills and the need for computer-programming aids for these machines.

1.1 WHAT IS NUMERICAL CONTROL

The definition of Numerical Control is as varied as its applications. Broadly speaking, Numerical Control may be defined as the control of a process by a servoed device or system to allow for performance of an ordered sequence of events along a path and at a rate predetermined by a group of alphameric instructions stored on a suitable medium. The most commonly used input medium is a 1-in-wide punched paper or Mylar tape running longitudinally.

Numerically controlled machine tools are guided by an automatic controller that responds to machining instructions contained in punched tape, punched cards or magnetic tape.

1.2 TYPE OF POSITIONING ARRANGEMENT

The control for a numerically controlled (NC) machine tool will be in general have either a point-to-point or a continuous-path type of positioning arrangement.

A point-to-point arrangement controls the positioning of a cutting tool or work table from one discrete point to another but does not control the path which it takes. This type of control arrangement is used in NC drilling machines.

In continuous-path arrangement the direction that the centre of the cutting tool takes is controlled at all times. Simultaneous control of one, two, three, or more axes of motion is possible. The example of such an arrangement is a NC contour milling machine.

1.3 NUMERICALLY CONTROLLED DRILLING MACHINES

The purpose of drilling is to obtain a hole in any given material.

Without numerical control several steps must be taken before a part can be made according to engineering drawing specification. First a production planer studies the engineering drawing and determines the operations necessary to complete the part in the shop. From this listing of the operations and specification of each operation, a tool designer plans the necessary jigs and fixtures.
If a drill jig is required consideration is given to the locating, or nesting, cf part during the operation as well as to the locating of the holes to be drilled.
When the workman prepares to drill a part, he locates the part in drill jig and clamps it into position. The

drill jig is manually moved across the fixed drill press table to locate the part under the drill press spindle. Some parts may require several drill jigs.

With numerically controlled point-to-point equipment, most of the operators control is replaced by automatic control. If the part requires a fixture, the operator clamps the workpiece to a fixture that has been positioned according to a designated setup point.

Following this, the control unit decodes instruction on punched tape to automatically move the drill column from point to point and at the end of an operation sequence, turn on a light to notify the operator to change the cutting tool.

1.4 NUMERICAL CONTROL TAPES

Full numerical control requires three types of information.

1. General instructions to the machine such as direction of motion, start, stop, feed, reverse, etc.
2. Numeric valuesof all operations. For example, length of travel, speed, etc.
3. Auxiliary functions, as tool change, coolent on or off, lubrication, and so forth.

All of this data will be encoded into punched tape. And this tape will be used to provide complete control for the machine tool.

1.5 COMPUTER PRODUCTION OF CONTROL TAPES

Although nearly all numerically controlled systems can be programmed manually with code tables, desk calculators and equipment for generating control tapes, this method is economically justified only when part configuration is very simple. Even then, the creation of a control medium can be rather involved. Computer is a means to provide efficient control tapes and cut down turn-around-time for manufacturing a part using an NC machine.

Computer programming systems like the APT (Automatic Programming Tool), AUTOSPOT, SNAP, EXAPT, II-CL etc are available which provide a means for producing control tapes for Numerical Controlled machines. All these NC programs aim to enable the user to prepare his part program with as little effort as possible by using a notation which is convenient for describing the engineering drawings. This notation is called as a partprogramming language. For detailed description of the programming system the reader is referred to the references (1-7).

Using the partprogramming language the user describes the workpiece and the machining sequences to the computer. The NC computer program does all the numerous computations necessary for producing the control tape for performing the necessary machining sequences.

1.6 WHY A SPECIAL PURPOSE PACKAGE

The programming systems just described are for general purpose NC machines.

For a multi-spindle drill, the number of holes to be drilled with all spindles operating, the number with one drill removed, and so on must be determined. This involves considering the pattern of holes and relative times of drilling and moving from one position to another. No standard program will do this and hence special software made to order is called for.

CHAPTER 2

THE SOFTWARE DESIGN

This chapter begins with the problem description and goes on to explain the evolution of a functional system level design of the software package.

2.1 PROBLEM DESCRIPTION

A software package has to be designed for computer preparation of control tapes for NC multispindle drilling machines INOCENTI and LAHR for drilling tube plates, support plates etc.

2.1.1 The Drilling Machines

Here the information relevant to the design only is included. More technical details can be found in Appendix A.

INOCENTI NC drilling centre has eight spindles. The spindles are arranged as shown in Figure 2.1. Centre to centre distance between spindles is fixed and cannot be adjusted.

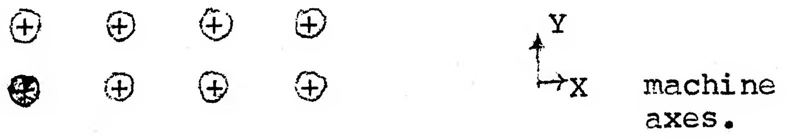


Figure 2.1

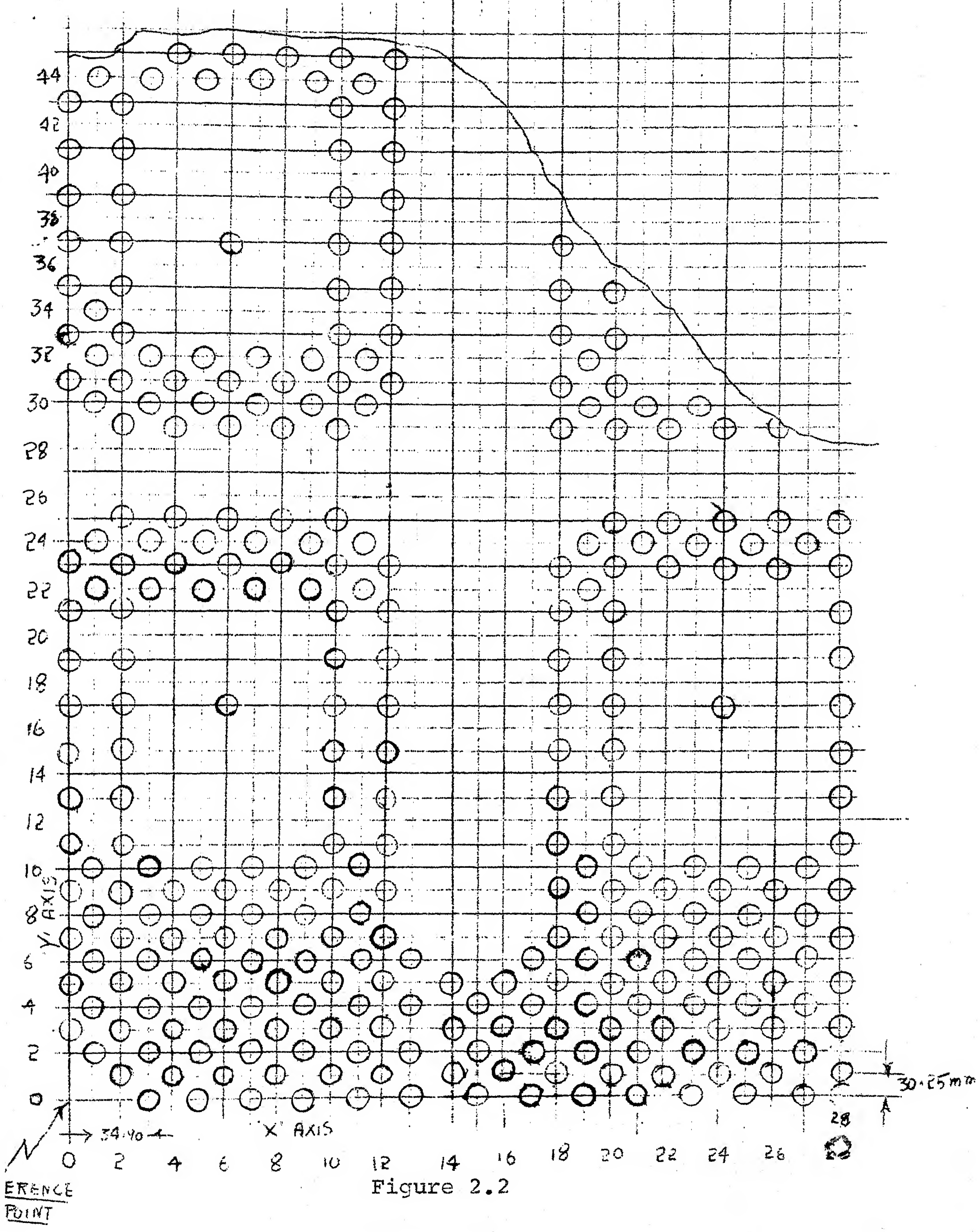
The spindle axis of the drilling machine is horizontal. It has a traverse area of 30Mx10M. The x-axis movement being along the horizontal and the y-axis movement along the vertical.

LAHR is a deep hole NC Gun drilling machine with a work area of about 2M x 2.75M. It has two spindles the distance between them being variable from 13.5 cm to 16.9 cm.

Both the drilling machines can have their drills accurately positioned anywhere in the work area by feeding the position information to the two-axis controller through punched paper tape input. Both drilling machines can be used with any combination of their spindles operating.

2.1.2 The Workpiece

Usually the workpiece has a very large number of holes to be drilled. A large majority of the hole centres are located on the intersection of perpendicular grid lines of an equispaced grid, the grid pitch being equal or a submultiple of the spindle pitch. This is apparently necessary for effective utilisation of the multispindle drilling capabilities of the machines. A part of an engineering drawing is shown in Figure 2.2 to make the picture clear.



44
42
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
Y' AXIS
'X' AXIS
34.40
30.25mm
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
ERENCE
POINT

Figure 2.2

2.1.3 Dimensioning for NC Drilling

Both incremental dimensioning and absolute dimensioning are in vogue. But the unique requirement for multispindle drilling as has already been mentioned is that the hole centres be located at the intersections of the perpendicular equispaced grid. This permits a very convenient method for specifying the location of the hole centres.

The grid pitch being known, the absolute location of the grid origin with respect to some reference point and the grid coordinates of the hole centre would locate it uniquely. This method of dimensioning is the standard practice for manual preparation of machine tool programs. Hence this type of engineering drawing could be assumed to be available to the partprogrammer also.

2.1.4 Drilling Philosophy

The total drilling time is roughly proportional to the number of discrete drill positionings. Hence in order to minimise drilling time, all holes which can be drilled with maximum number of drillbits in position must be drilled first and then with the next lower combination of spindles operating and so on, until all holes are covered. Hence programming involves pattern matching - the drill pattern has to be systematically matched with the hole patterns.

2.2 SYSTEM OBJECTIVES

The following system objectives for the software package emerged from the study of the problem described in the last section.

(1) An easy to learn input language (part-programming language) be provided for describing the work to be drilled and to specify the associated technological data.

(2) Provide adequate error diagnostics and debugging facilities for the partprogrammer.

(3) Simulate the drilling machine movement to prepare a machine tool program for drilling holes described by the partprogram.

(4) Provide computer print out of drilling pattern with each spindle set.

(5) Convert the machine tool program to punched paper tape output to EIA specifications for the concerned machine.

(6) Provision for batch processing of part-programs.

(7) Must have adequate software mobility and there should be scope for continuous improvement of facilities provided by the package.

2.3 SYSTEM DESIGN

A system has been designed to meet the objectives set in the last section, the essential features of which are explained here with the help of a flowchart.

2.3.1 The Computer

The software has been written for IBM 7044, but can be easily adopted to any other computer with a FORTRAN compiler and adequate core storage. As very few machine dependant features have been used, the modifications needed are minimal.

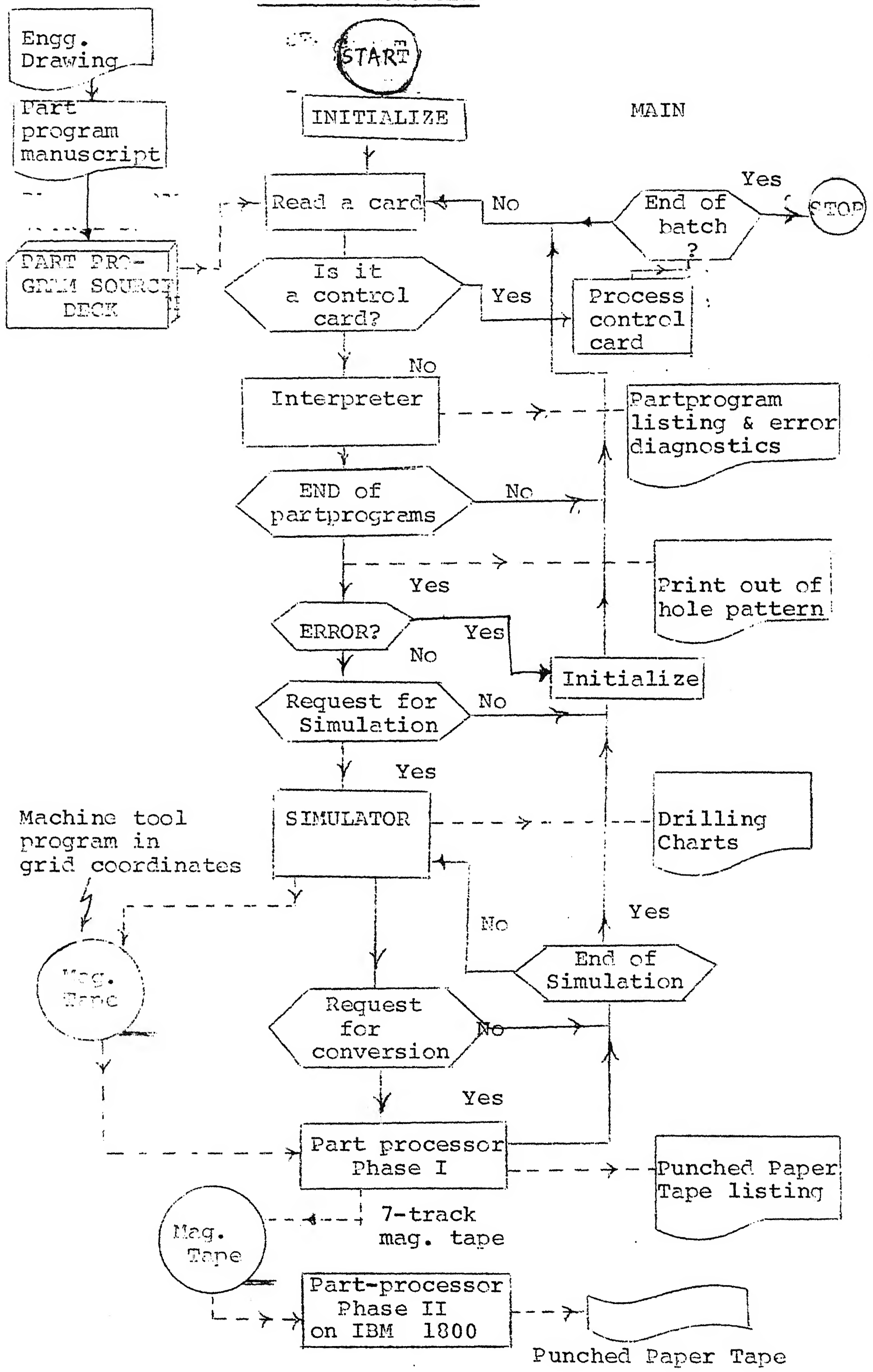
2.3.2 Source Language

Except for a few routines which are in assembly language of IBM 7044, the source language of the package is FORTRAN IV. This gives the software adequate mobility.

2.3.3 Partprogram

The pattern of holes to be drilled on a part is specified with the help of a symbolic language specially designed for this purpose. The symbolic instructions are punched one-per-card. The associated technological data appearing on control cards. The design of the input language is discussed in Chapter 3.

SYSTEM FLOWCHART



2.3.4 Functional Division

The package has been functionally divided into four modules. This task oriented division facilitates incorporation of modification and improvements in one module without necessitating extensive changes in the package as a whole. Each module is built up of one or more subroutines.

(1) MAIN : This is the supervising section ensuring adequate flow through the various sections and catering for correct branching to other sections. Once processing through the other three sections (modules) has been completed, the control is returned to main, which initializes the input of a subsequent partprogram. The control cards are also processed by this section. The control parameters are saved for later use.

(2) The Interpreter : The interpreter combines the translation and execution of the sequence of input symbolic pattern description language statements. The processing is purely sequential. The output of the interpreter is the image of the hole centres defined by the symbolic statements in the internal memory of the interpreter.

(3) The Simulator : This section determines the drilling sequence for each spindle set by performing a controlled simulation of the drilling process of the

concerned machines with the core image built up by the interpreter as the virtual workpiece.

(4) Postprocessor: The task of the postprocessor is to convert the drilling sequence data output of the simulator to paper tape conforming to EIA specifications for the concerned machines.

The postprocessor has two phases. Phase-I for conversion to EIA format with certain amount of editing. The Phase-II is a program written for IBM 1800 computer for converting the magnetic tape output of Phase-I to paper tape. Two phases are necessitated due to the absence of paper tape punch on IBM 7044 computer.

A detailed description of the working of the various modules appears in succeeding chapters.

CHAPTER 3

THE PATTERN DESCRIPTION LANGUAGE

The design of an input language for describing the hole patterns is described in this chapter.

3.1 TYPES OF INPUT DATA

The input data to the NC computer program falls into two distinct categories and needs to be processed in different ways.

3.1.1 Technological Data

The technological data like the selection of the drilling machine, the spindle combination, the tool description, output mode etc is provided through control cards. These allow the user to specify the above data and also output options. The format and the description of the legal control cards is given in Appendix C.

3.1.2 Pattern Description

The hole patterns are described using a symbolic language with a fixed format. This gives the required terseness and still retain the intelligibility of the partprogram without making the software package unduly bulky.

The repertoire of the input language is evolved through a study of the patterns of holes normally expected to be encountered. The holes are specified by a combination of commands to locate and delete holes. The commands locating or deleting holes are interpreted sequentially in the order of their appearance. The commands are punched one-per-card.

3.2 THE SYMBOLIC LANGUAGE

The selection of the symbolic instruction for pattern definition is explained in this section. A more detailed description of each instruction with examples is given in Appendix C

3.2.1 Instruction Format

1 5	6	7 12		14 72	
Label Field		Opcode		Variable field	Seq. field

Label Field: may contain a name by which other instructions can refer to the instruction labelled.

Opcode Field: This field must contain the symbolic opcode for the instruction.

Variable Field: The field contains the necessary arguments for the instruction. ',' (comma) separates the arguments. The first 'blank' character terminates the variable field.

3.2.2 The Instructions

3.2.2.1 Grid Definition:

GRID $x_M, y_M, x_{int}, y_{int}, x_n, y_n$

It is found that for describing the location of the hole centres the natural choice is to define the grid on which the holes are located and then individually or in groups specify the location of the hole centres in grid coordinates, as this would involve minimum computation by the partprogrammer, as the drawings are grid oriented. (See Figure 2.2).

x_M, y_M are coordinates of the origin of the grid with respect to machine adjusted origin. x_{int}, y_{int} are the horizontal and the vertical grid spacings respectively, x_n, y_n defines the size of the grid plane in number of vertical and horizontal grid lines.

3.2.2. Definition of Basic Hole Patterns: The basic hole centre patterns on the grid are either isolated points or evenly spaced points on vertical or horizontal grid lines (See Figure 2.2).

(1) POINTG x_g, y_g

The above statement defines a hole centre at x_g, y_g on the grid with respect to the origin.

(2) LINEH x_g, y_g, d, x_m

LINEH defines equispaced hole centres on a horizontal grid line. x_g, y_g is the location of the left most hole centre, 'd' is the centre to centre to distance between adjacent holes and x_m is the x-coordinate of the last hole centre.

(3) LINEV x_g, y_g, d, y_m.

LINEV defines equispaced hole centres on vertical grid line. It must be noted that the instructions avoid the need for counting the number of hole centres and hence decrease the possibility of a human error.

3.2.2.3 Repetition of Basic Patterns: The total hole pattern on a part can be divided into subpatterns, which have either a repetition of horizontal points pattern or a vertical points pattern. Hence it should be sufficient to just describe the horizontal or vertical points pattern and specify the repetition. This is achieved by a REPEAT statement.

REPEAT 'label',d,n

All LINEH and LINEV statements are duplicated n times upto and including the statement with the 'label'. While duplicating LINEH statements, y_g is incremented by an amount 'd' everytime and while duplicating LINEV statements, x_g is incremented by an amount 'd' every time.

3.2.2.4 Shift of Origin: Some times it may be convenient to define some hole centres with respect to an origin other than the absolute grid origin.

ORIGIN x,y

This statement shifts the origin to x,y on the grid. All the statements following this will refer to this origin until another ORIGIN statement changes it.

3.2.2.5 Deletion of Defined Hole Centres: In a majority of cases it is easier to define patterns, by first defining simple repetitive patterns and then selectively deleting certain holes

DELETE ON DELETE OFF

To effect the deletions the same statements which define the hole centres are used by just changing the mode of the program. If in one section of a partprogram hole deletions are to be commanded, the program is switched to 'delete' mode by a DELETE ON statement and switched back to hole centre definition by a DELETE OFF statement at the end of the section.

Often it is required to delete a rectangular area of holes on the grid. This is facilitated by a CLEAR instruction.

CLEAR x_1,y_1,x_M,y_M.

(Refer to Appendix C for details).

3.2.2.6 <u>Symmetry Considerations:</u> If a pattern is symmetrical about any of its central (horizontal or vertical) grid lines it is only necessary to describe one half (one quarter if it is symmetrical about both the central grid lines) and describe the symmetry by MIRROR statement

```
MIRROR [UP]
       (DOWN]
       [LEFT]
       [RIGHT]
```

3.2.3 <u>Definition and use of Subpatterns</u>

The total pattern of holes or a portion can be sometimes broken up into a repetition of a single pattern with some translation and/or inversion. From this the concept of subpartprogram follows by which one needs to define the pattern only once and then generate the rest of the patterns by calling it by the name given to it and appending appropriate modifiers. The modifiers define the necessary translation and/or inversions.

```
Block name   BLOCK  x,y  ¦   ENDB   Blockname.
```

BLOCK and ENDB are the delimiter statements for a subpartprogram. Block name is the unique name given to the block pattern being defined. x,y gives the grid dimension for the patterns.

For example,

```
TEST    BLOCK   20,20
        REPEAT  NEXT,1,21
NEXT    LINEH   0,0,1,20
        CLEAR   8,1,19,19
        DELETE  ON
        LINEV   4,4,2,18
        ENDB    TEST
```

Previously defined patterns can be used to build a new pattern.

The point pattern can be called up by its symbolic name and logically ANDed or ORed to the grid after an inversion and/or translation by giving suitable modifers.

```
e.g.     TEST     INVERT,OR,20,30
```

(For other options refer to Appendix C.)

With the vocabulary so far described the input language has a powerful point pattern definition capability. The details of the computer program for interpreting this is explained in the next chapter.

CHAPTER 4
THE INTERPRETER

This explains the programming techniques adopted in the interpreter section of the package.

4.1 DATA STRUCTURE

It has been already mentioned that the hole centres occur at the intersection of the horizontal and vertical grid lines. The diameter of all the holes defined in a partprogram is the same. The partprogram specifies, on which of the grid intersections hole centres occur. This information has to be stored in some convenient form for further use.

An immediate choice would be to represent the grid by a binary matrix with one word per 'intersection'. But the storage requirement for a reasonable sized grid of say 200 x 100 is 20K words. Unless complex core swapping techniques are adopted such core storage requirements cannot be accommodated.

The alternative is to associate a bit with each intersection of the grid lines. A '1' bit indicating a hole at that intersection and a '0' bit indicating absence of a hole. This arrangement makes the storage problem immediately manageable. However, FORTRAN does not provide bit manipulation. The actual arragement of the bit pattern representing the grid is discussed elsewhere.

4.2 LEXICAL ANALYSIS

Lexical analysis is to check whether a symbolic instruction conforms to the rules of the language or not.

4.2.1 Seperation into Syntactic Units

Card images are supplied in unpacked form to the lexical analysis routine. It collects the syntactic units from the various fields of the card image. The variable field is broken up into subfields with ',' (comma) as the break character. It determines for each syntactic unit, if it has a numeric or a non-numeric header character and packs this information into a 'type' word with the bit in the position corresponding to the sequence number of the syntactic unit in the card image indicating the type of the header character. A '1' bit representing a numeric header and a '0' bit a non-numeric header character. It trys to convert all syntactic units in the variablefield with a numeric character header to binary representation. In case there is a non-numeric character in such a syntactic unit, conversion is aborted and an error diagnostic results.

e.g.

```
1LOOP     REPEAT    FINI, M1, 20
```

'Type' word |///////|10001|

No. of syntactic units = 5.

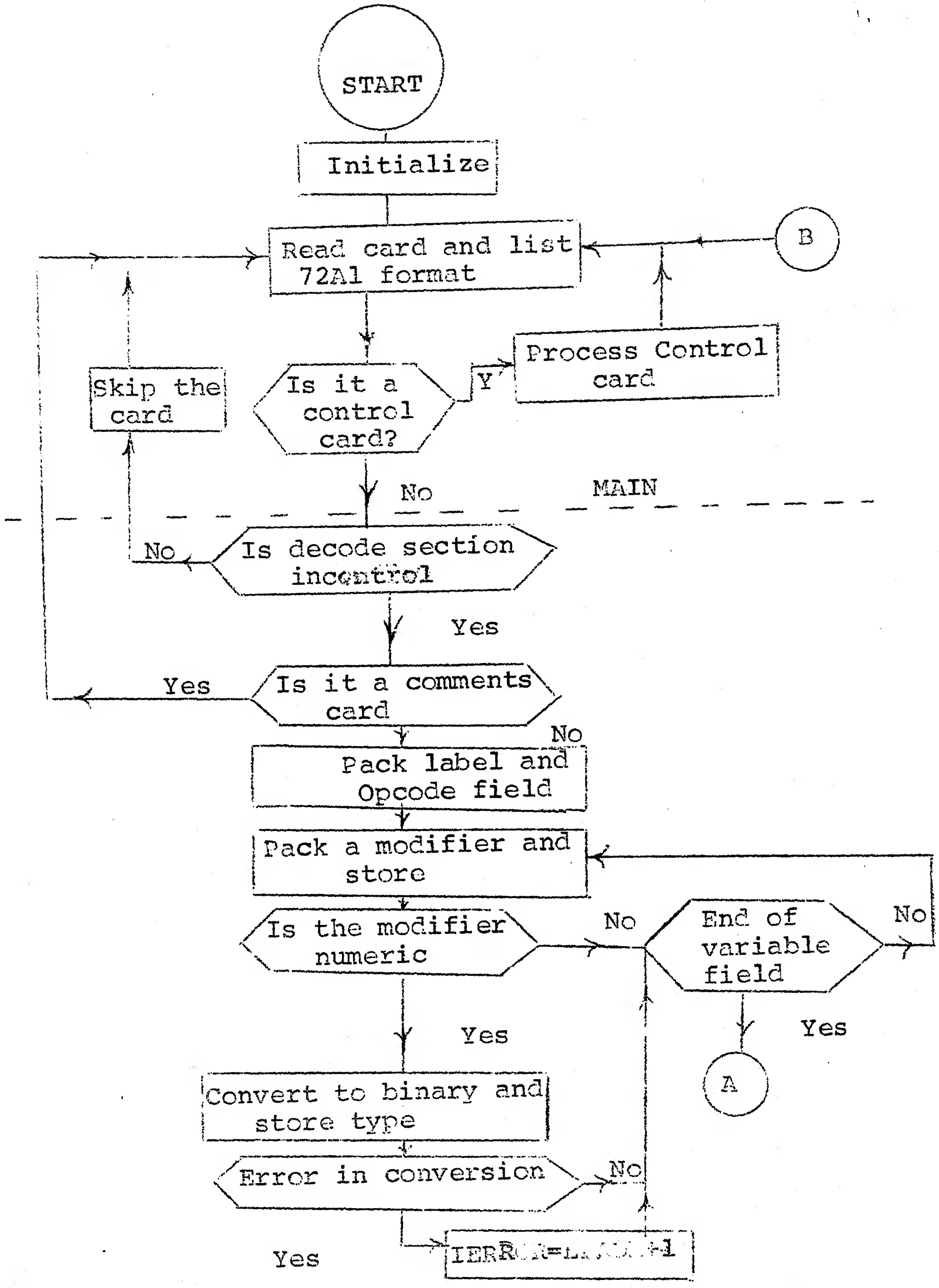
4.2.2 Opcode Table

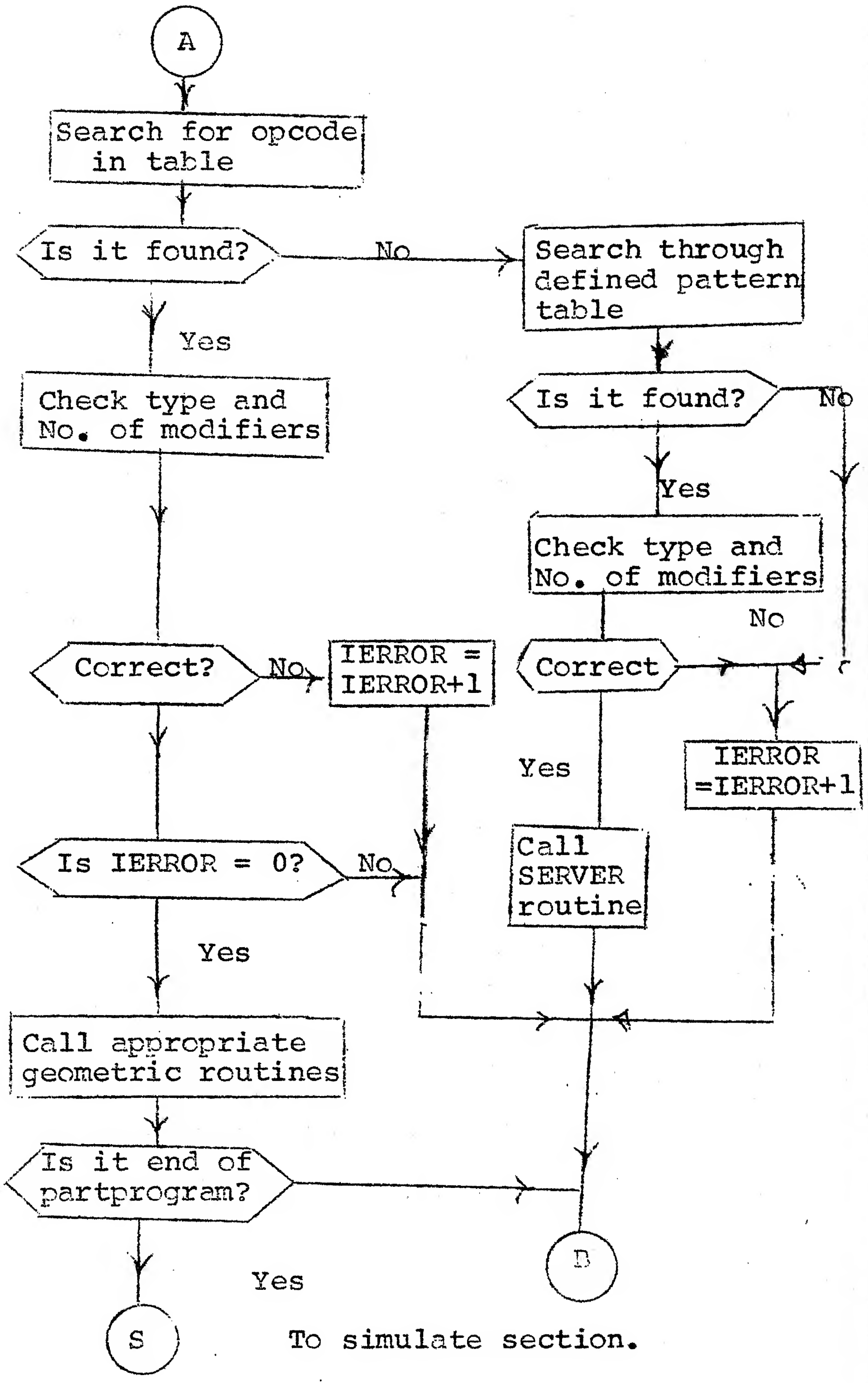
The information regarding each type of instruction must be stored in some convenient way in the interpreter to check whether or not an instruction in a partprogram is valid, and if valid what action is to be taken. This is achieved by means of a legal opcodes table.

The table is a linear array with three word entries for each of the different instructions in the pattern definition language. The entries for the instructions are ordered according to their frequency of use in a typical partprogram.

The table arrangement is as follows:

Word Index	Opcode	No. of syntactic units	Type word
1	LINEH	6	/////|111100
4	POINTG	4	////////|1100
7	LINEV	6	/////|111100
10	REPEAT	5	//////|11000
.	.	.	.
.	.	.	.
.	.	.	.


START
Initialize
Read card and list 72A1 format
B
Is it a control card?
Y
Process Control card
Skip the card
No
MAIN
Is decode section incontrol
No
Yes
Is it a comments card
Yes
No
Pack label and Opcode field
Pack a modifier and store
Is the modifier numeric
No
End of variable field
No
Yes
A
Yes
Convert to binary and store type
Error in conversion
No
Yes
IERROR=IERROR+1



A
Search for opcode in table
Is it found?
No
Search through defined pattern table
Yes
Check type and No. of modifiers
Is it found?
No
Yes
Check type and No. of modifiers
No
Correct?
No
IERROR = IERROR+1
Correct
IERROR =IERROR+1
Yes
Is IERROR = 0?
No
Call SERVER routine
Yes
Call appropriate geometric routines
Is it end of partprogram?
B
Yes
S
To simulate section.

Some checks for partprogramming rules violation are carried out have not been shown in the flowchart.

Of the three words, the first word contains the symbolic code name for the instruction. The second word contains the number of syntactic units it must have. The last word is the 'type' word for the instruction.

4.2.3 Execution of an Instruction

A linear search is made through the opcode table to locate the symbolic code name. On locating it,a check is carried out to verify whether the instruction has the correct number and type of syntactic units. The index of the instruction in the table directs a call to the appropriate geometric routine for executing the instruction. In case of an error a message to that effect is printed and the link to geometric routines is broken.

4.2.4 Interpreter Flowchart

A flow chart of interpreter section is included to further the understanding of its working. It may be observed from the flow chart that in case of an error in interpreting or executing an instruction further processing of the partprogram through the other sections is discontinued but processing through the interpreter section itself continues.

4.3 GEOMETRIC ROUTINES

Each instruction in a partprogram is in effect an indirect call to one of the so-called geometric routines. The action of these routines in association with each instruction is explained in this section.

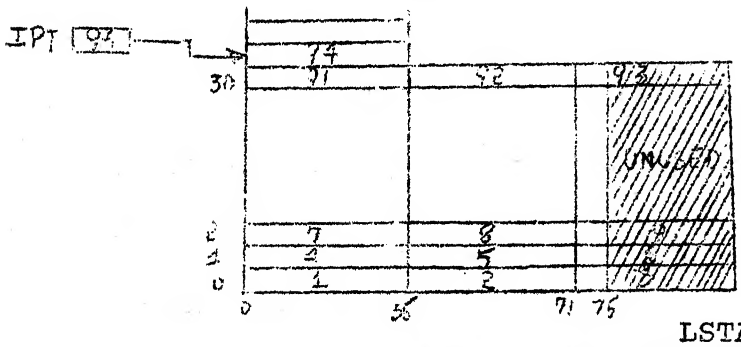
4.3.1 GRID instruction

The interpreter has as its internal memory a one dimensioned FORTRAN variable of five thousand words (STORE(5000)).

On encountering a GRID instruction requisite memory for the grid is cleared and allocated. The internal representation of a grid is a packed matrix of '0' and '1' bits. This is explained below with the help of an example.

Example

GRID 0,0,1130,2200,75,30



```
                              LSTAT(1) = 1
         Status variable      LSTAT(2) = 75
                              LSTAT(3) = 30
```

Parameters of the GRID operation are saved and the global variable LSTAT updated to contain the pointer to location and dimensions of the grid. Use of the global status variable LSTAT is an interesting programming trick adopted which permits an instruction in a part-program and in a subpartprogram to be handled logically in the same manner. IPT is a pointer to the first free location in the internal memory.

4.3.2 DELETE instruction

On a DELETE ON instruction the 'delete' flag is set 'on'. The delete flag is set 'off' on encountering a DELETE OFF.

4.3.3 REPEAT instruction

The variable field of the REPEAT instruction is saved in a 'repeat' status variable.

On encountering an instruction with a label matching the one contained in the last REPEAT instruction, the instruction is executed and the program is restored to its 'non repeat' status.

An unterminated or nested REPEAT block results in an error diagnostic and the link to geometric routines being disabled.

4.3.4 POINTG instruction

This is the most rudimentary operation in the input language. As the base address of the grid and the dimensions of the grid are known, the word address and bit position representing the intersection specified by the instruction can be calculated. Then this bit is made '1' or '0' depending on the 'delete' flag.

4.3.5 LINEH instruction

LINEH operation defines evenly spaced holes on a horizontal grid line. As the internal representation for a horizontal grid intersectionpoints is a row of the matrix, the following scheme has been adopted. The bit pattern defined by the instruction is generated in a buffer. Then the buffer is logically ORed to the correct row of the matrix.

More often than not LINEH instruction appears in a REPEAT block. By constructing the bit pattern in a buffer instead of directly on to the matrix, it becomes necessary only to repeatedly perform a logical 'OR' operation onto the matrix, the row index being controlled by the 'repeat' parameters.

In case the 'delete' flag is 'on' the constructed bit pattern in the buffer is logically complemented and logically ANDed to the correct row of the matrix.

4.3.6 System Defined Macros

4.3.6.1 LINEV instruction: LINEV is an example of the macro capability of the input language.

LINEV is a system defined macro which expands as follows.

```
    REPEAT   LBL1,$d_1$,n           REPEAT LBL1,$d_1$,n
        .                               .
        .                               .
        .                               .
                                    Save 'Repeat' status
    LINEV    $x_1$,$y_1$,$d_2$,$y_2$      REPEAT LBLS,$d_2$,$(y_1-y_2)/d_2+1$
                               LBLS LINEH  $x_1$,$y_1$,$d_1$,$d_1$ x (n-1)
                                    restore 'repeat' status

        .                               .
        .                               .
        .                               .
LBL1                           LBL1
```

4.3.6.2 <u>CLEAR instruction:</u> CLEAR instruction is again a system defined macro.

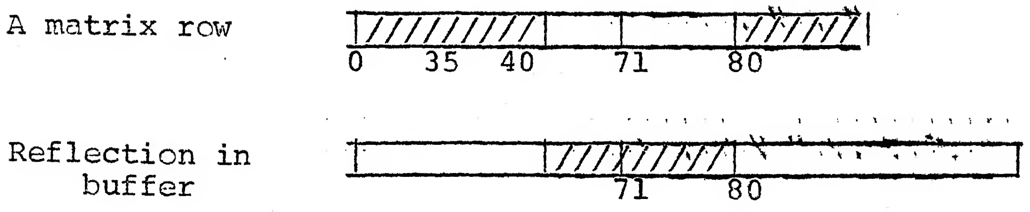
```
        CLEAR $x_1$,$y_1$,$x_2$,$y_2$  expands as shown below
              save 'repeat' status
              save 'delete' status
              REPEAT LBLS,1,$y_2$,$y_1$+1
        LBLS  LINEH $x_1$,$y_1$,1,$x_2$
              Restore 'repeat' status
              Restore 'delete' status.
```

4.3.7 <u>MIRROR instruction</u>

Mirroring about the horizontal central axes of the grid is achieved by generating the correct indexing for copying the rows of the matrix.

Mirroring about the vertical central axes is not so straight forward because of the way the grid is internally represented.

Considering a horizontal grid line with an example the method is illustrated below for a MIRROR RIGHT instruction.

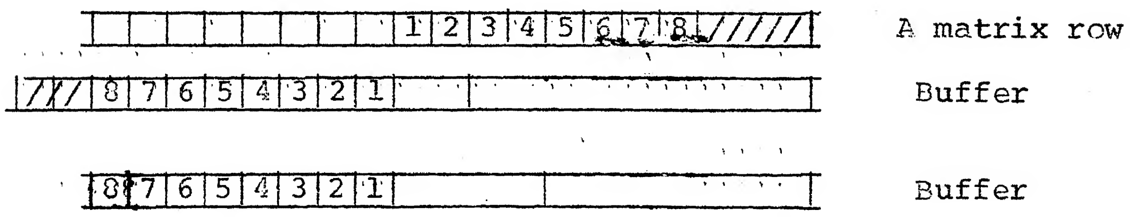


The reflection of the left half of the matrix row is built bit by bit in a buffer and then logically ORed to the matrix. This process is repeated for all the rows of the bit matrix.

The MIRROR LEFT instruction is executed in a slightly different way which is again illustrated below by an example.

<u>Example</u>

Word length = 6 bits



First the bit positions are reversed in each word and stored in the buffer as shown. Then the whole buffer is shifted right and logically ORed to the matrix row. This process is repeated over all the matrix rows.

4.4 SUBPARTPROGRAMS

The mechanism of subpartprograms can be implemented in two different ways. One method is to store the subpartprogram and execute it whenever it is called up by its name. Not only is this costly in terms of execution time, but also the purely sequential nature of interpretation will have to be sacrificed.

Hence the alternative, which is to generate the pattern as it is being defined and store it in a convenient form and use it whenever it is called has been adopted.

4.4.1 Definition of Patterns BLOCK and ENDB instructions

BLOCK and ENDB instructions are the delimiters of a subpartprogram.

As soon as the BLOCK instruction is encountered the status of the partprogram including the 'repeat' status and 'delete' status is saved. The requisite memory for the pattern is cleared and allocated for the pattern and a four word entry is appended to the 'defined' patterns table. The global status variable LSTAT is also updated to contain the pointer to location of the allotted storage and dimensions of the pattern. IPT is updated.

The 'defined' patterns table is also linearly arranged as shown below.

Word index	Symbolic pattern name	Pointer to storage	DIMENSIONS	
			x_1	y_1
1	↓	↓	↓	↓
5				
9				

When the ENDB instruction is encountered the program status is restored, including the global status variable LSTAT. The pattern defined has been generated and tucked away in the internal memory.

4.4.2 <u>Calling of Defined Patterns</u>

If the symbolic opcode is not found in the opcode table then a search through the 'defined' patterns table is carried out. When the pattern name is located in the 'defined' patterns table the pointer to storage entry directs to the location of the pattern. The pattern which is stored in bit matrix form is brought into a buffer row by row shifted into position and directly logically ORed ord ANDed depending on the modifier, to the grid defined by the global status variable LSTAT.

In case there is an inversing modifier present (TURNIN and ROTATE) a reversal of bit pattern preceeds shifting. As this process is independent of where the pattern is being called from, previously defined pattern can be called for defining another pattern.

CHAPTER 5

THE SIMULATOR

This chapter is devoted to the explanation of how the mechanism of multispindle drilling is simulated to produce a drill program.

5.1 INPUT TO THE SIMULATOR

The input to the simulator is the master bit matrix constructed by interpretation of the partprogram.

Three planes which are bit matrices of the same dimensions as the grid bit matrix are created in the internal memory after clearing the bit patterns created by the subpartprograms. These can be called as the work planes for the simulator. The master bit pattern is copied into work plane number one.

5.2 INITIAL SELECTION OF SPINDLES

Intune with the drilling philosophy explained in Chapter 2, first as many holes as possible must be covered with the largest feasible set of spindles operating.

Spindle sets which are feasible for a particular work piece are those which have centre to centre distance between spindles compatible with the grid spacing. For example INCOCENTI NC drilling centre has two rows of four spindles each. In case the distance between vertical grid lines is not equal to,or a submultiple of the distance between spindle centres on the two adjacent rows, it would not be possible to drill with two rows of spindles operating.

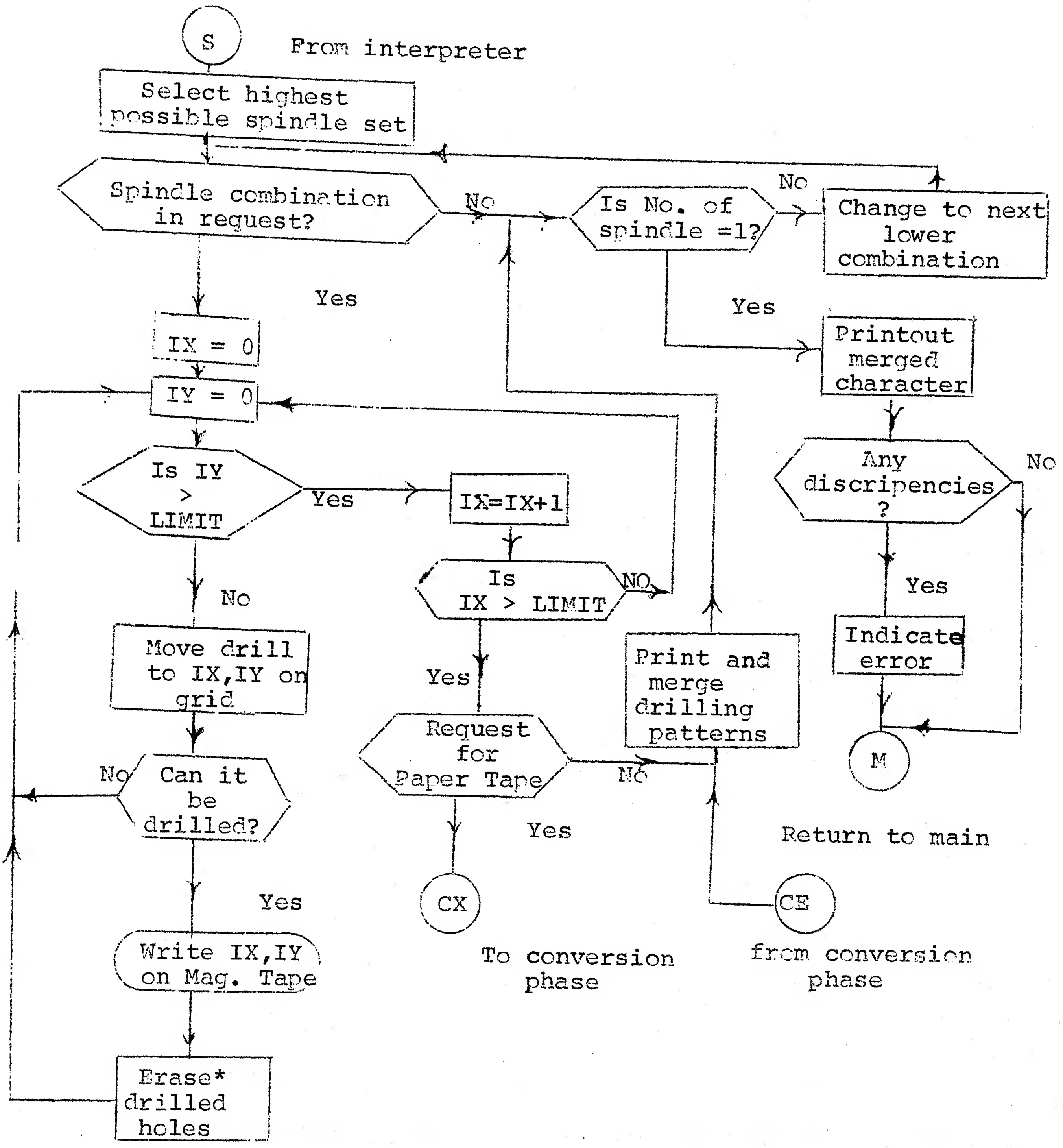
5.3 DRILLING MACHINE REPRESENTATION

The spindle pattern of the selected combination is constructed in a 36-bit word,a '1' bit representing an operating spindle and '0' bit representing absence of a spindle. This representation for the drilling machine is compatible with the bit matrix representation of the workpiece.

With this completed,the drilling process is just a matter of matching of bit patterns. For positioning the drill anywhere on the grid the drill pattern is copied into a double-word and appropriately shifted and the word address of the x,y grid intersection in the work plane number one is calculated.

The double-word bit pattern is matched with the bit matrix words whose address has been calculated. If there are two rowsof spindlesit is matched with two rows of the bit matrix. The match of all '1' bits corresponding to the spindles with the '1' bits corresponding to the hole centres indicates that in this drill

SIMULATOR FLOW CHART



*Refer to text

position the workpiece can be drilled. On this the matching '1' bits in the 'bit' matrix are erased and the corresponding bits in the second working plane are turned on and the x,y grid coordinates are recorded on tape. In case of an unsuccessful match no action is taken. The second working plane will have the pattern of holes covered by the spindle set.

5.4 DRILL MOVEMENT ALGORITHM

When using NC drills, the sequence of drilling holes plays an important part in reducing the total drilling time for a workpiece.

With reference to INOCENTI and LAHR NC drilling machines; the controllers take less time for positioning the drill if the direction of approach for positioning is positive, i.e. the increments in x and y coordinates is positive. This would mean the sequence of drilling must be such that as far as possible the direction of approach is positive.

The number of holes that can be covered with a spindle set and the corresponding set of drill positions is determined by a sweeping traverse of the working plane number one by the simulated drill explained in the last section. The traversing is to be such that maximum successful positionings result. It was found that a uniform sweep of the drill pattern parallel to the vertical grid lines resulted in a very good solution, irrespective of the pattern characteristics. This sweep left the holes which were not covered by the spindle set in clusters at the edges of the pattern and not scattered all over. This clustering makes it possible to cover the majority of these with the next lower combination. The traversing is explained in Figure 5.1 and in the accompanying flowchart. This simple traverse also provides a drilling sequence with a positive positioning approach.

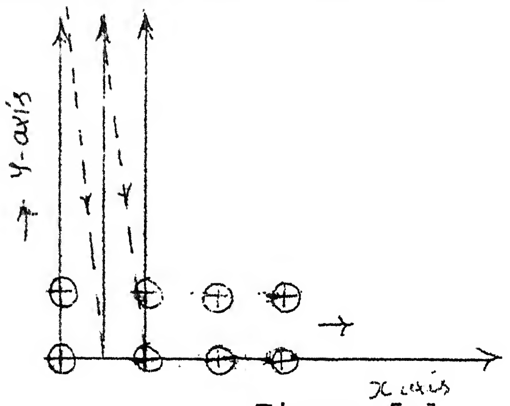


Figure 5.1

It is not possible to extract any quantitative information from the hole pattern to optimize the sweep, because a global look at the pattern cannot be taken. The attepmts which could only be intuitive resulted in no better solution than with the simple uniform sweep.

5.5 CHANGE OF SPINDLE SET

Once simulation is completed with the one spindle set the spindle pattern word is modified to the next lower combination in the list of requests given on the /$MACH card by the user and the simulation repeated to to determine the drilling sequence for the new spindle set and so on until the requests are exhausted.

5.6 DRILLING CHARTS

As already mentioned working plane number two will contain at end of simulation with each spindle set, the hole pattern covered with that set. This bit matrix is unpacked substituting a '0' character for '1' bit (a hole) and a 'blank' character for a '0' bit and printed out. If the width of the chart exceeds the printer capability, the chart is split into several parts and printed out. This chart aids in tape program checkout.

5.6.1 Merged Drilling Chart

This drilling chart is merger of all the individual drilling charts for the various spindle sets. This is prepared by merging the charts on the magnetic tape after simulation with each spindle set and printing out at the end. This chart would be very useful to the operator of the NC drill for a visual check when the actual drilling is in progress.

CHAPTER 6

POST-PROCESSOR

Any computer program designed to prepare control tapes for NC machine tools must consider both the geometry of the component to be machined and the dynamics and other particular requirements of the machine tool system. In present practice, this is achieved, for any one system, either by using a single special purpose program written to suit the particular machine tool or by sharing the generation of control tapes between two distinct programs- a general-purpose processor and a post-processor. The processor produces a general solution for the partprogram. The post-processor converts this solution to punched tape in the format specified for the concerned machine controller.

Sembelance of such a division exists in this NC computer program also, though the post-processor is an integral part of the software package.

6.1 POST-PROCESSOR PHASE I

The sequence of drill positionsfor each spindle set in terms of grid coordinates written on magnetic tape by the simulator forms the input to the phase.

These coordinates are converted to absolute machine coordinates using the information saved from the GRID instruction (absolute coordinates of the grid origin with respect to the machine origin and the horizontal and vertical grid pitch.).

6.1.1 Tape Format

The tape for the two machines is to conform to EIA standard RS-273A. This is a word address variable block format. The essential featuresof this format is given in Appendix B.

Each positioning command for the drill is a block of characters on the tape. The block being terminated by a special EOB character. Each block consists of several words of information. As the commands are sequential, some words may be omitted from a block. The machine controller interprets this to mean no change in the state of the machine with respect to the omitted word. For example, if in a positioning command, one of the coordinates of the tool position is the same as for the immediately preceding command, the coordinate word may be omitted from the block. This makes the length of the block variable.

6.1.2 Editing

The absolute coordinate position data is converted sequentially to a block of charactersto the format described above and written into a 120 characters fixed length buffer. When the buffer becomes full,it is written out on 7 track magnetic tape in BCD mode as an unblockd twenty word record without a control word and a writing into a new buffer is initiated. The conversion at present is made using TAPE 99 feature available on IBM 7044. But this may as well be done using a physical tape,though it will be slightly slower. The leading zeros in the dimension words in the blocks cannot be inserted in FORTRAN. These appear as blank characters on the magnetic tape. The tool change and setup commands (from SAFPOS and SETUP instructions) are inserted in their proper positions in the sequence of commands. At the end of the simulation any half filled buffer is emptied out filling it with EOR characters. A printout of what would be punched on tape is provided by Phase I.

6.2 PHASE II

The Phase II is a program for converting the character stream on magnetic tape to papertape. This phase is on the IBM 1800 computer which has a paper tape punch as a peripheral.

This phase need consider the input from the magnetic tape as just a character stream, because all necessary formatting has already been carried out in Phase I.

A code conversion from the BCD magnetic tape code to the paper tape code is necessary. This is achieved using a conversion table. The octal representation of each BCD character is used as the address to the word containing the correct paper tape code for the character in the table. This is a very fast conversion technique as no search need be made through the table. All 'blank' characters are converted to '0' characters(for inserting leading zeros). One physical record on the magnetic tape is read at a time, converted and punched on to the paper tape and the process repeated until the EOF mark is encountered on the magnetic tape.

CHAPTER 7

CONCLUSION

A computer programming system for automatic programming of multispindle numerically controlled drilling machines has been developed. This is an economic means for producing efficient and accurate drilling sequence control tapes which cut down turn around time per workpiece considerably.

Though it is doubtful whether a criteria for theoretical optimum drilling sequence could be arrived at, it is observed that even if such optimisation were to be possible the total saving would not be more than 1-2% of total drilling time.

The modularity of the package and the functional simplicity of the simulator makes it possible to introduce into the package any technical recommendation that arise from user experience to produce more efficient control tapes.

An offshoot of the development of this package is a powerful symbolic point patterns definition language. The apparent generality of this language makes it possible to use it as a tool for pattern matching problems other than the one for which it has been designed for.

REFERENCES

1. Leslie W.H.P, Editor "Numerical Control Programming Languages" Proceedings International IFIP/IFAC PROLAMAT Conference, ROME, 1969.

2. Leslie,"Numerical Control User's Hand Book",McGraw-Hill,1970.

3. Oleston, "Numcrical Control", Wiley Interscience, 1970.

4. Robert B.Thornhill, "Engineering Graphics and Numerical Control",McGraw-Hill, 1967.

5. IITRI, "APT Partprogramming",McGraw-Hill, 1970

6. "1620/1311 AUTOSPOT III Application Description" IBM Application Program

7. "AUTO PROPS II- IBM Application Program Bulletin,1962.

APPENDIX A

Only selected data from the specifications has been included in this appendix.

1. INOCENTI NC DRILLING CENTRE

Control System: DATEX NC system made by Conrac Corporation, U.S.A.

System Specifications

Data Input: EIA standard BCD 1-inch, 8 track, perforated tape. Rotary digiswitches in mannual numeric.

Reading and Rewind Speed: 300 characters/sec (photoelectric)

Traverse Rates:	2200 mm/min first approach
	30 mm/min second approach
	0.6 mm/min creep
Resolution:	0.01 mm lineal (fine)
	0.04 mm lineal (Coarse)
Displays (5):	X actual (seven digits plus sign)
	Y and R-actual (6 digits plus sign)
	Sequence (3 digits).
	Tool (2 digits)

<u>Display Maximum:</u>	99999.99 mm	X
	9999.99 mm	Y and R
	999	sequence
	99	tool.

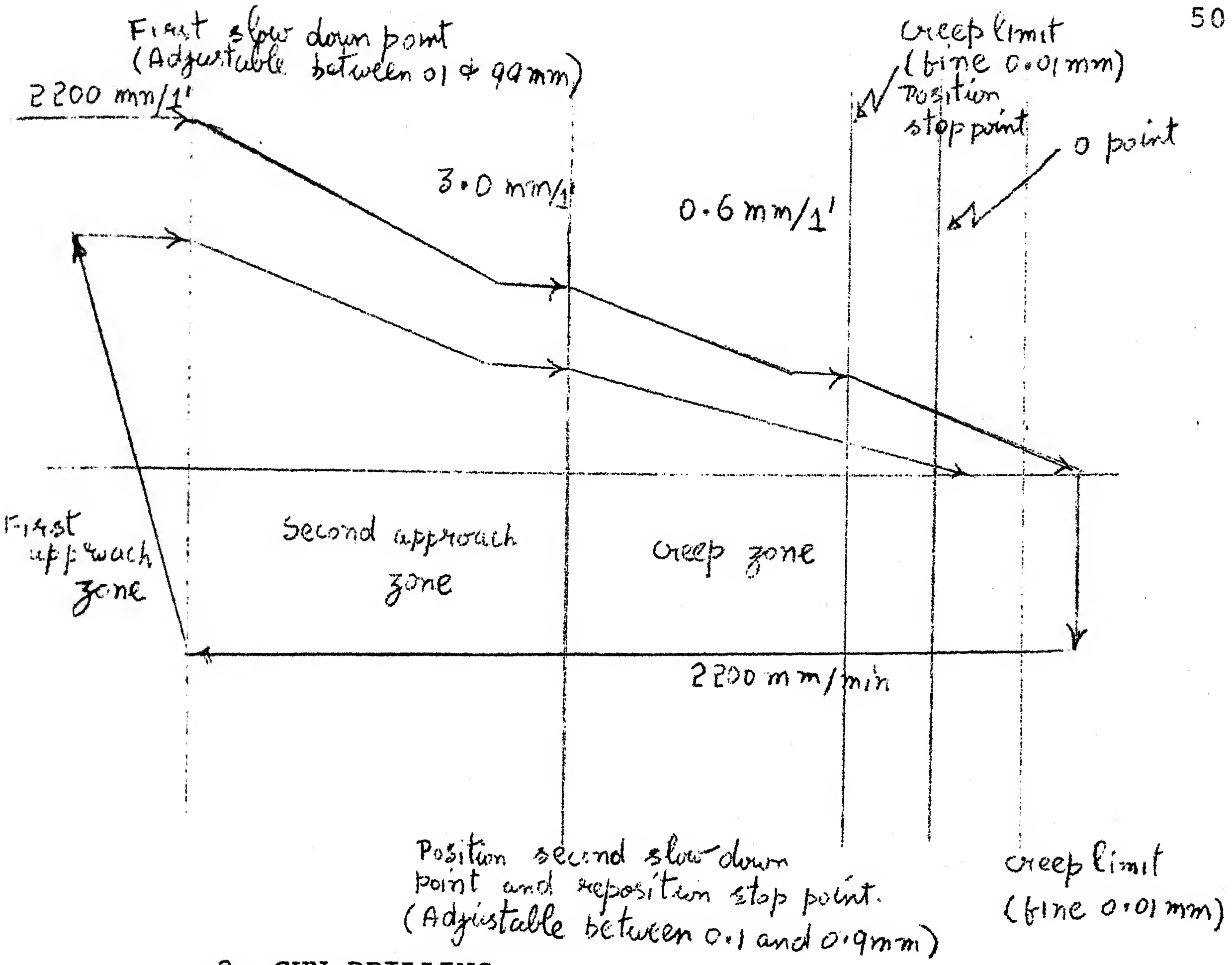
<u>System Control:</u> Point to point positioning. Straight line or angle milling control. Two axis simultaneous control.

<u>Offset:</u> Capable of full-range zero offset for all axes.

<u>Operating Modes:</u> Coarse positioning
Accurate positioning positive approach
Auto repositioning
Accurate positioning negative approach.

<u>Auto Repositioning:</u>

3-Speed initial, one speed return and two speed repositioning approaches are illustrated. If the machine is located within one or more limits when positioning begins, the higher speed(s) will not be utilised.



2. <u>GUN DRILLING</u>

Many years ago gun drilling was the term applied to drilling gun barrels. Now it is applied to those drilling operations which require an exceedingly deep hole with a close tolerance.

A special type of drill is required for this type of drilling. It is two fluted, usually. carbide-tipped, and has a hole running the full length of the drill. Oil is pumped through the hole during the drilling operation to keep the drill cool and in cutting. Gun drilling may be performed by rotating either the drill or the workpiece. The purpose of gun drilling is to obtain precise holes in relatively high production. Reductions of upto 50% of cost have been obtained by utilising numerical control for gun drilling, in addition, much greater accuracy is achieved.

3. <u>WARNER and SWASEY-LAHR GUN DRILLING MACHINE</u>

(a) <u>Specification</u>

. spindle area

vertical	6'
horizontal	8'

. No. of spindles 2

. Spindle stroke (max.) 40"

. Numerical control system : Mark century 7532/3. Absolute positioning 2-axis, $\pm$ programming, mirror image.

(b) <u>Basic Control Features</u>

. Basic control - 2-axis positioning, format x $\pm$ 43, y $\pm$ 43 metric

. 300 characters/sec tape reader

. RS 244 tape code : Sprocket hole verification, parity check, parity over ric

. 0.001 and 0.01 millimeter incremental feed.

. 0.008 mm absolute positioning accuracy.

(c) <u>Director</u>

Address information from the data input selects the motions to be positioned and provides the command signals for directing those motions. The position information provided by the data input is used to set up a position command singal. The position feedback units provide an indicationof the machines position which is compared with the commanded position. Any difference in correspondence, which represents the position error will provide a signal to the servo drive control to position the motion.

When the position error exceeds a preset amount (adjustable 3 mm to 14mm with 1 mm gearing), the error limit function will hold up the command pulse train until the error becomes less than the set value.

APPENDIX B

1. Character code for Numerical Control perforated tape EIA standard RS-244-A.

87654 321	Track Number
`     .  o`	1
`     . o `	2
`   o . oo`	3
`     .o  `	4
`   o .o o`	5
`   o .oo `	6
`     .ooo`	7
`    o.   `	8
`   oo.  o`	9
`  o  .   `	0
` oo  .  o`	a
` oo  . o `	b
` ooo . oo`	c
` oo  .o  `	d
` ooo .o o`	e
` ooo .oo `	f
` oo  .ooo`	g
` oo o.   `	h
` oooo.  o`	i
` o o .  o`	j
` o o . o `	k
` o   . oo`	l
` o o .o  `	m
` o   .o o`	n
` o   .oo `	o
` o o .ooo`	p
` o oo.   `	q
` o  o.  o`	r
`  oo . o `	s
`  o  . oo`	t
`  oo .o  `	u
`  o  .o o`	v
`  o  .oo `	w
`  oo .ooo`	x
`  ooo.   `	y
`  o o.  o`	z

87654 321	Track Number
`oo o.  oo`	.
` ooo.  oo`	,
` oo  .  o`	/
`ooo  .   `	+
`o    .   `	−
`    o.oo `	&
`o oo . oo`	%
` ooo .oo `	Tab
`o    .   `	Car,ret. or End of Block
`oooo.ooo`	Delete
`    o. oo`	End of Record
`   o .   `	Space
`  o o.  o`	Back Space
`oooo .o  `	Upper Case
`oooo .  o`	Lower Case
`     .   `	Blank Tape
	Virgin Tape

2. TAPE FORMAT

The control systems use a word-address format. On the tape, alphabetic codes are assigned to each coordinate and function word. These letters appear on the tape as the first character of the word and are followed by the numeric position of the word. Each word is ended with a tab-code. Each group of words (block) which make up a machine sequence is ended with a EOB character. EOB is also the first character in tape. For any particular block only those words which give new information need appear. When a word is missing from a block, the tab-codes must still appear until the last useful word in the block is reached.

The tab and End-of-Block codes have dual roles since, in addition to their control system use, they also serve on the tape-preparation typewriter equipment to space out the words horizontally on the typescript (tabulation) and vertically because the EOB code is also the 'New Line' code for the typewriter.

BLOCK FORMAT

Word	
n x x x Tab	Sequence Number - This word begins with n followed by a 3-digit sequence number and a Tab code. This sequence number must be the first word in each block. All 3 digits must be coded even though zero. (i.e., n001).
x + or - x x .x x x . x x	x-command word. The letter x, a sign (+ or -) and 7 decimal digits (INOCENTI) defines the absolute horizontal movement of the column with respect to the manually selected x-offset. The decimal point is is just shown for reference, it is not coded on the tape. All seven digits must be coded even though zero (.e., x+00110.50)
Y + or - x x x x . x x Tab	Y-command word. Details same as for x-command word.
t : x x tab	Tool number word - the letter t and 2 decimal digits define the tool number. This number is displayed on the display panel and reminds the operator of the tool to be used.
m x x Tab EOB	Miscelleneous function word - The letter m and two decimal digits. This controls machine function like programme stop (m00), End of program (M02) etc.

In each block, the different words must appear in the sequence shown above.

APPENDIX C

DRILLPAL USER'S MANUAL

1. MOTIVATION

DRILLPAL is a software package for the computer preparation of control tapes for multiple spindle NC drilling machines INOCENTI and LAHR.

If one is conversant with the manual preparation of control tapes and the rudiments of cartesian rectangular coordinate system he is all set to become a DRILLPAL user.

If a computer is to prepare the control tape, it must be provided with all information a partprogrammer would need for manual preparation of control tapes.

The partprogrammer is supplied with the engineering drawing of the part to be processed, specifications of the drilling machine to be used and drilling recommendations, if any. He has to pass on this information to the computer. For precisely doing this, the DRILLPAL provides him with a partprogramming language.

In the following pages the user is introduced to the elegant repertoire of the language, what it can do and what it can't, and the rules to be observed in using it.

2. SOME DEFINITIONS

(a) PART: A term used for a mechanical component which has to be produced by an NC machine.

(b) LANGUAGE: The sum of the agreed forms of statements which are accepted by a system for legitimate commands.

(c) PARTPROGRAM: A sequence of instructions which describes the work which has to be done on a part, in the form required by a computer under the control of an NC computer program (e.g. in a source language such as one provided by DRILLPAL).

3. DRILLPAL NC COMPUTER PROGRAM

This NC computer program allows batch processing of partprograms.

The partprogram is to be punched on 80 column IBM card in a format which is to be described later. A card in a partprogram deck can either be a control card, or a card containing a valid geometric statement. Comments can be inserted in the deck by a comments card. If a card in the deck is none of the above an error will invariably result.

Execution of the partprogram is purely sequential. Partprogram is not stored but is executed instruction by instruction as it occurs.

4. CONTROL CARDS

Control cards are characterised by a '/' punch in the first column and '$' punch in the second column. They are used to dellineate decks, select drilling machines, drill bit, mode of output desired etc.

Valid control cards, their functions and formats are described below.

(a) /$PART Card

Format

```
 1      8          16
/$PART  partname   [TAPE],;tape name ø     any text.
                   [NOTAPE]
```

This card is meant for delineation of partprograms, /$PART card must be the first card in every partprogram. Absence of this card will lead to skipping of the whole partprogram deck.

Partname: Partname is any combination of one through six BCD characters. This will appear in the computer printout as a reference to the part programmed.

TAPE Option: This option specifies that punched paper tape to control the NC drill is to be produced. Listing of the paper tape will be included in the print out.

NOTAPE Option: This option specifies that punched paper tape is not to be produced. In this case only the grid coordinates for point to point positioning of the drill will be included in the printout.

Tape name: Tape name is the name of the user magnetic tape reel on which the intermediate output is to be produced. (This will have to be mounted on C4 on IBM 7044.)

Machine tool program written on the magnetic tape is to be later converted to punched paper tape using IBM 1800.

Any text appearing on this card will be saved and reproduced as the header of the drilling chart etc.

e.g.

```
1       8        16                                      72
/$PART  BPLATE   TAPE,'941UȻBOILER PLATE DRG.K037
```

Note: ' (quote mark) must be present before the tape name.

(b) /$MACH CARD : Format

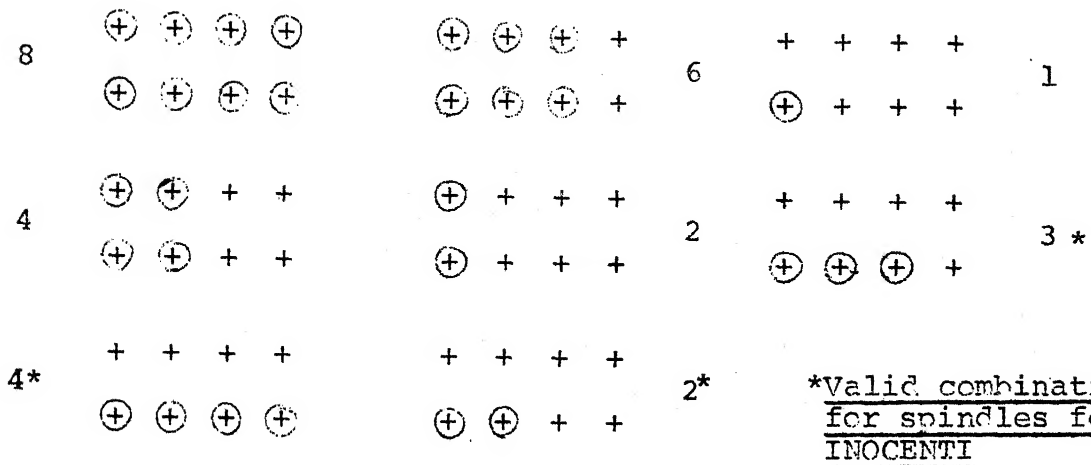
1	8	16	
/$MACH	machine name	$n_1, n_2, \ldots,$ Ȼ	Any comments

This card is used to select the drilling machine and to specify the combination of drills to be tried out.

The simulator always selects the maximum number of spindles first and then change to next lower number of spindles specified and so on.

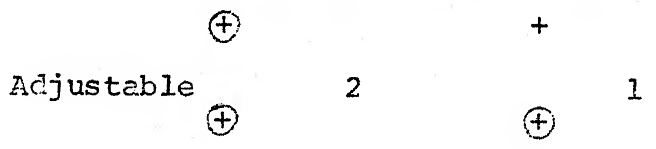
<u>Machine name</u> : INOCENTI or LAHR

<u>$n_1, n_2, \ldots$</u> : gives the combination of drill spindles to be tried out.



*Valid combinations for spindles for INOCENTI

Valid number of spindles for LAHR Gun Drill is 2 and 1



In the absence of the /$MACH card only core image generated by the partprogram will be printed out.

e.g.

```
1         8       16                                         72
/$MACH    LAHR    1,2Ø    USE  INCOCENTI IF POSSIBLE
```

*Three combinations can be given only when the vertical spindle pitch is not computable with the centre to centre distance between horizontal grid lines.

(c) /$DBIT CARD

Format

1	8	16	72
/$DBIT	Drillbit size	Any further specifications	

Drillbit Size: It can be specified in any manner desired. It can be any combination of one thru six BCD characters.

This card is not very significant to the NC computer program. This is because the present version of the DRILLPAL assumes that all holes to be drilled at the hole centres defined in a partprogram are of the same diameter. It does not check for the diameter of the hole. It is solely the responsibility of the part-programmer to select the drillbit.

The drillbit size and any text appearing on this card will appear in the drilling chart printouts.

Only one /$DBIT must be present in a partprogram deck. Absence of this card will not cause any error.

e.g.

1	8	16	72
/$DBIT	25.4mm	COOLANT IS TO BE USED	

(d) /$PRNT CARD

Format

```
 1                                        72
/$PRNT    any comments
```

As soon as this card is encountered a printout of the hole centres defined by the preceeding geometric statements is produced. If it is in the middle of a BLOCK definition a print-out of the block pattern so far defined is produced.

This is a utility for partprogram debugging.

(e) /$OVER CARD

The format

```
 1                                        72
/$OVER        Any Comments
```

This card indicates the end of the batch of part-programs. This card must be present.

5. COMMENTS CARD

A * in the first column indicates that the card contains comments. This card will be listed with the source. It exists solelY for the convenience of the programmer and does not effect execution of the program.

6. DRILLPAL REPERTOIRE

(a) The Format: Fixed format with a certain amount of flexibility is the style.

Card Format

1 5	6	7 12	13	14 72	80
Label field		Operation Field		Variable Field Comments	Seq. Field

(1) Label Field: Thid field may be either blank or contain a label.

A label is any combination of one thru five alphameric characters. The first character must be an alphabet. Blank characters are ignored in the label field.

(2) Operation Field: This field must contain a legal opcode defined 'BLOCK name' Blank characters are ignored.

(3) Variable Field: This field must contain the necessary arguments for the opcode or 'Block' name on the operation field. The arguments are separated by commas. The first blank character ends the variable field and the rest is treated as comments.

DRILLPAL checks for each operation, the variable field to see if it contains the required number and type of arguments.

(b) Data Statements

(1) GRID : Grid Definition Statement: A majority of the hole centres have to be located on a equispaced rectangular grid on the workpiece for effectively utilising the multiple spindle drilling capability. DRILLPAL has been made-to-order specifically for producing control tape

(2) <u>SAFPOS</u>

Tool change position

```
6 7          12  14
_________________________
  SAFPOS         x,y
```

x,y is the safe position for tool change. Whenever a tool change has to be made the drill carriage will move to this position.

Units	100ths of a mm for INOCENTI
in integers	1000ths of a mm for LAHR

Default Option: SAFPOS 0,0

(3) <u>SETUP</u>

```
5 7          12   14
_________________________
  SETUP           x,y
```

x,y gives the setup point for work alignment. Units same as for SAFPOS.

Default option: SETUP 0,0

(c) <u>GEOMETRIC STATEMENTS</u>

Geometric statements define hole centres on the grid. Units for all arguments are in grid coordinates unless otherwise specified.

(1) <u>ORIGIN</u>

```
7           14
____________________
ORIGIN      x,y
```

Sometimes it may be convenient to describe some centres with respect to an origin other than the GRID origin. This can be achieved with a shift of origin by an ORIGIN statement.

All the Geometric statements following this statement will refer to this origin until another ORIGIN statement changes it.

(2) <u>POINTG</u>

7	12	14
POINTG		x_g, y_g

POINTG defines a hole centre on the grid at x_g, y_g with respect to the origin.

(3) <u>LINEH</u>

7	14
LINEH	x_g, y_g, d, x_m

LINEH defines equispaced hole centres occuring on a horizontal grid line.

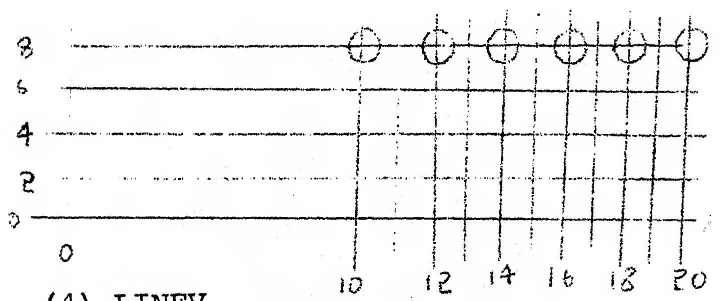
(1) x_g, y_g is the location of the left most hole centre.

(2) 'd' is the centre to centre distance of adjacent holes.

(3) x_m is the x-coordinate of the last hole centre.

e.g.

LINEH 10,8,2,20



(4) LINEV

7	14
LINEV	x_g, y_g, d, y_m

LINEV defines equispaced hole centres occuring on a vertical grid line.

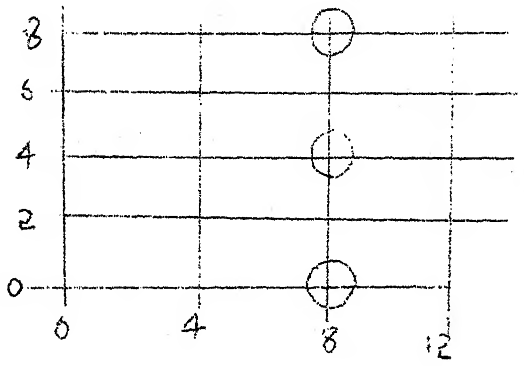
(1) x_g, y_g is the location of the bottom most hole centre.

(2) 'd' is the vertical centre to centre distance between adjacent holes.

(3) 'y_m' the y-coordinate of the top most hole centre.

e.g.

LINEV 8,0,4,8



(5) <u>REPEAT</u>

```
     7           12
    REPEAT         Label,d,n
```

All LINEH and LINEV will be duplicated n times upto and including the statement with 'label'. While duplicating LINEH statement, y_g will be incremented by an amount d everytime.

e.g.

```
         REPEAT    LOOP,3,3
         LINEV     10,3,2,11
         LINEH     8,7,3,17
    LOOP POINTG    15,14
```

will be expanded as follows in the partgrogram.

```
          LINEV    10,3,2,11
          LINEV    13,3,2,11
          LINEV    16,3,2,11
          LINEH    8,7,3,17
          LINEH    8,10,3,17
          LINEH    8,13,3,17
    LOOP  POINTG   15,14
```

(6) DELETE

```
7        12 14
DELETE      [ON]
            [OFF]
```

It may be sometimes convenient to define a group of hole centres and then selectively delete some of the holes. This can be achieved by the DELETE statement.

(1) ON Option: ON option causes all geometric statements following this statement to define hole centre deletions. If a command to delete an undefined hole centre is given, it does not cause an error.

(2) OFF Option: OFF option brings the interpreter to its normal mode of hole centre definition.

(7) MIRROR

```
7        12   14
MIRROR        [UP]
              [DOWN]
              [LEFT]
              [RIGHT]
```

MIRROR statement causes reflection of hole centres defined, about the central axes. The reflection is controlled by the option in the variable field.

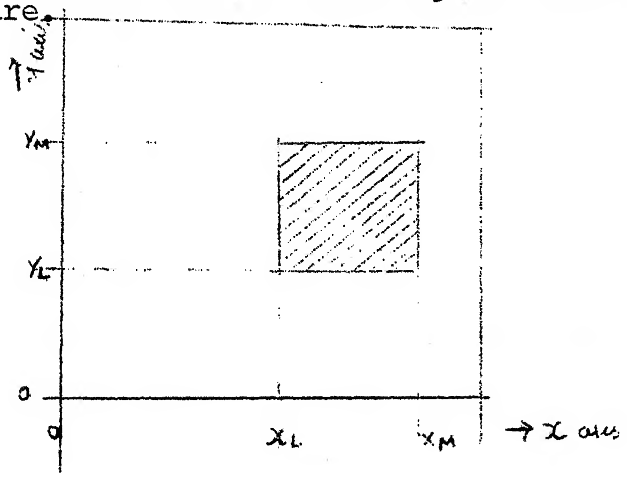
$\substack{\text{[UP]}\\\text{[DOWN]}}$ Option: The hole centres defined $\substack{\text{below}\\\text{above}}$ the horizontal central grid line to be mirrored $\substack{\text{up}\\\text{down}}$ about the central grid line.

$\substack{\text{[LEFT]}\\\text{[RIGHT]}}$ Option - The hole centres defined to the $\substack{\text{right}\\\text{left}}$ of the vertical central grid line to be mirrored $\substack{\text{left}\\\text{right}}$ about the central grid line.

(8) CLEAR

```
1          14
CLEAR      X_L,Y_L,X_M,Y_M
```

This statement clears a rectangular area of hole as shown in Figure.



The holes on the bounding lines of the rectangle are also cleared.

(d) PATTERN DEFINITION

If a pattern of hole centres can be broken up to some translation and/or inversion of some sub-patterns, it would be convenient to define these sub-patterns and copy these sub-patternson to the main grid plane with the required translation and/or inversion. This would reduce the part-programming effort and also reduce the errors in part-programming.

For precisely doing the above DRILLPAL provides a means for writing a subpartprogram within a main part-program. 'BLOCK' statement and the ENDB are the delimiters for the subpartprogram.

(1) BLOCK

```
     5   7       12   14
Block   BLOCK         x,y
name
```

(2) ENDB

```
  7          14
ENDB       Block name
```

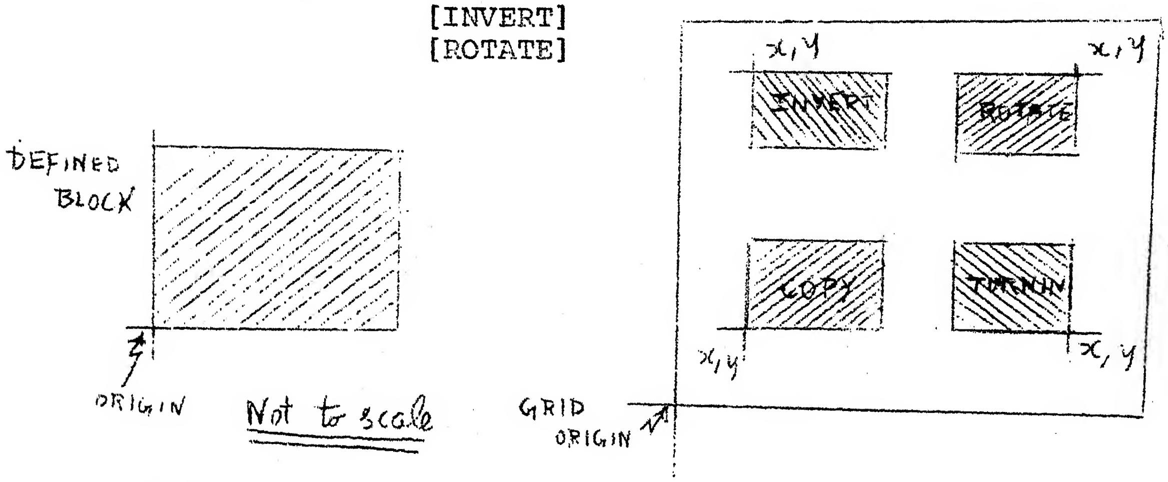
(1) Block Name: Block name is a unique name given to the pattern being defined. It may contain 1 thru 5 BCD characters. The first character must be an alphabet.

(2) x,y defines the size of the grid for the pattern.

All the geometric statements explained until now and also pr-viously defined 'BLOCK' patterns can be used in a 'BLOCK' definition. But nesting of 'BLOCK' definitions is not permitted. All the statements between the BLOCK statement and ENDB statement form the body of the pattern. DELETE status and REPEAT parameters will be saved and restored.

The following is the format for using defined 'Blocks'.

```
7      12 14
BLOCK     [COPY]    , [AND] , x,y
name      [TURNIN]    [OR]
          [INVERT]
          [ROTATE]
```



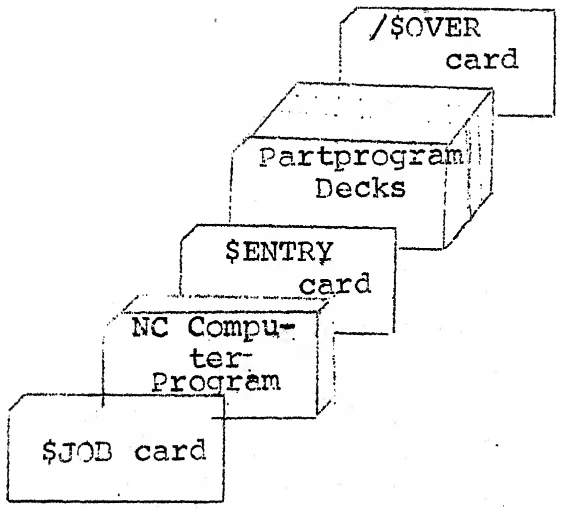
[COPY] [TURNIN] [INVERT] [ROTATE]	option is explained in the figure.
[AND] [OR]	option logically ANDs / ORs the pattern to the grid.

x,y is the location of pattern origin on the grid.

(7) A Sample Partprogram

```
/$PART SAMPLE TAPE, !941U   THIS IS FINAL DEBUG RUN

/$MACH INOCENTI8,4,2,1

* A TEST PART PROGRAM   DATED 6/4/72

/$DBIT 25.4MM    TWIST DRILL

      GRID     0,0,1745,3025,144,82
      SAFPOS   -75000,15125
LBOT  LINEH    11,0,2,59
      LINEH    10,1,2,62
      LINEH    9,2,2,63
      LINEH    6,3,2,64
      LINEH    5,4,2,65
      LINEH    2,5,2,64
RBOT  LINEH    81,0,2,135
      LINEH    78,1,2,136
      LINEH    77,2,2,139
      LINEH    76,3,2,142
      LINEH    75,4,2,143
ENDRB LINEH    76,5,2,144
      REPEAT   LOWER,2,18
      LINEH    1,6,2,143
LOWER LINEH    0,7,2,144
* CLEAR THE UNWANTED HOLES
      CLEAR    66,6,74,30
      CLEAR   443,31,99,31
      CLEAR    60,32,79,41
      CLEAR    96,32,98,41
      CLEAR    99,38,144,41
      MIRROR   UP
      END      SAMPLE END OF PART PROGRAM
      /$OVER
```

(8) <u>Deck Composition</u>



EE-1972-M-NAY-SOF